Data Analysis and Visualization in R (IN2339)

Exercise Session 3 - Tidy Data & Combining Tables

Felix Brechtmann, Vicente Yépez, Žiga Avsec, Julien Gagneur

Section 00 - Getting ready

```
library(data.table)
library(magrittr)
library(tidyr)
```

Section 01 - Tidy Data Warm Up

1. Examine the dataset `AirPassengers`. Which of the following is true:

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

   a. `AirPassengers` is tidy data: it has one year for each row.
   b. `AirPassengers` is not tidy: we need at least one column with a character vector.
   c. `AirPassengers` is not tidy: it is a matrix instead of a data frame.
   d. `AirPassengers` is not tidy: to be tidy we would have to wrangle it to have three columns (year, month and value), then each passenger count would have a row.

```
# d
```

2. Examine the dataset `ChickWeight`. Which of the following is true:

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

a. `ChickWeight` is not tidy: each chick has more than one row.
b. `ChickWeight` is tidy: each observation (a weight) is represented by one row. The chick from which this measurement came is one of the variables.
c. `ChickWeight` is not tidy: we are missing the year column.
d. `ChickWeight` is tidy: it is stored in a data frame.

```
# b
```

3. Examine the dataset `spanish_vowels`. Is the data set tidy?

```
##             label rep frequency1 frequency2
##   1:   p01-male-a   1   615.4477   1230.806
##   2:   p01-male-a   2   644.6112   1281.965
##   3:   p01-male-a   3   607.9174   1247.960
##   4:   p01-male-e   1   476.9079   1612.076
##   5:   p01-male-e   2   457.2205   1839.456
##  ---
## 746: p50-female-o   2   577.1894   1310.138
## 747: p50-female-o   3   545.5014   1214.094
## 748: p50-female-u   1   405.7645   1491.935
## 749: p50-female-u   2   458.0345   1141.513
## 750: p50-female-u   3   457.4308   1181.657
```

```
# No it is not tidy because the label contains multiple values (participant, sex, vowel).
```

4. The `example_product_data.csv` file describes the number of times a person bought product "a" and "b". Load the file into a `data.table`.

```
product_dt <- fread('extdata/example_product_data.csv')
product_dt
```

```
##            name producta productb
## 1:     John Doe       NA       12
## 2:    Marry Doe        3        1
## 3: John Johnson        5        1
```

5. Transform `product_dt` into a long fromat using data.table commands.

```
long_dt <- melt(product_dt, id.vars = "name", value.name = "n", variable.name = "product")
long_dt
```

```
##            name  product  n
## 1:     John Doe producta NA
## 2:    Marry Doe producta  3
## 3: John Johnson producta  5
## 4:     John Doe productb 12
## 5:    Marry Doe productb  1
## 6: John Johnson productb  1
```

6. Transform the table from the long format back into a wide format. Check that it is equal to the original data.table.

```
dcast(long_dt, ... ~ product)
```

```
##            name producta productb
## 1:     John Doe        NA       12
## 2: John Johnson         5        1
## 3:    Marry Doe         3        1
```

Section 02 - Weather dataset

1. Read in the weather dataset `weather.txt` as a `data.table`.

```
messy_dt <- fread("extdata/weather.txt")
head(messy_dt)
```

```
##             id year month element d1  d2  d3 d4  d5 d6 d7 d8 d9 d10 d11 d12 d13
## 1: MX000017004 2010     1    TMAX NA  NA  NA NA  NA NA NA NA NA  NA  NA  NA  NA
## 2: MX000017004 2010     1    TMIN NA  NA  NA NA  NA NA NA NA NA  NA  NA  NA  NA
## 3: MX000017004 2010     2    TMAX NA 273 241 NA  NA NA NA NA NA  NA 297  NA  NA
## 4: MX000017004 2010     2    TMIN NA 144 144 NA  NA NA NA NA NA  NA 134  NA  NA
## 5: MX000017004 2010     3    TMAX NA  NA  NA NA 321 NA NA NA NA 345  NA  NA  NA
## 6: MX000017004 2010     3    TMIN NA  NA  NA NA 142 NA NA NA NA 168  NA  NA  NA
##    d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30 d31
## 1:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 278  NA
## 2:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 145  NA
## 3:  NA  NA  NA  NA  NA  NA  NA  NA  NA 299  NA  NA  NA  NA  NA  NA  NA
## 4:  NA  NA  NA  NA  NA  NA  NA  NA  NA 107  NA  NA  NA  NA  NA  NA  NA
## 5:  NA  NA 311  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 6:  NA  NA 176  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

2. Why is this dataset messy?

3. How would a tidy version of it look like? Do not give the code, only describe how the tidy table would look like.

```
## Why is it messy?
## 1. Variables are stored as columns (days)
## 2. A single entity is scattered across many cells (date)
## 3. Element column is not a variable.
##
## Tidy version: id, date, tmin, tmax
```

4. Create a tidy version of the weather dataset.

```
## wide -> long
dt <- melt(messy_dt, id.vars = c("id", "year", "month", "element"),
           variable.name = "day",
           value.name = 'temp')

# you can ignore the warning message
dt[, day := as.integer(gsub("d", "", day))]
```

```
dt[, date := paste(year, month, day, sep = "-")] # option using paste
dt[, c("year", "month", "day") := NULL] # remove reduntant columns
# alternatively one can use unite from the tidyr package
# dt <- unite(dt, "date", c("year", "month", "day"), sep = "-", remove = TRUE)

dt[, element := tolower(element)] # TMAX -> tmax
dt <- dcast(dt, ... ~ element, value.var = "temp") # long -> wide

dt <- dt[!(is.na(tmax) & is.na(tmin))] # remove entries with both NA values,
                                       # na.omit(dt) would also do the job

head(dt)
```

```
##             id       date tmax tmin
## 1: MX000017004  2010-1-30  278  145
## 2: MX000017004 2010-10-14  295  130
## 3: MX000017004 2010-10-15  287  105
## 4: MX000017004 2010-10-28  312  150
## 5: MX000017004  2010-10-5  270  140
## 6: MX000017004  2010-10-7  281  129
```

Section 03 -Scattered data across many files

The `baby-names` folder contains 258 csv-files (`1999.girl.csv`, `1999.boy.csv` , ...) which store name frequencies for a particular year and sex.

1. Create a list containing all file paths in the folder.

```
files <- list.files("extdata/baby-names", full.names = TRUE)

# See one file
head(fread(files[1]))
```

```
##       name  percent
## 1:    John 0.081541
## 2: William 0.080511
## 3:   James 0.050057
## 4: Charles 0.045167
## 5:  George 0.043292
## 6:   Frank 0.027380
```

2. Name the list entries with the basename of the corresponding file path.

```
# name the list elements by the filenames
names(files) <- basename(files)
```

3. Read in the data from all files into one table. *Hint*: when you read many files and gather them into one table, be sure to add a column that identifies each file. `rbindlist()`

```
# read all files at once into a list of data.tables
tables <- lapply(files, fread)

# bind all tables into one using rbindlist,
# keeping the list names (the filenames) as an id column.
dt <- rbindlist(tables, idcol = 'filename')
```

4. Is the data tidy? If not, tidy it up.

```
# The data is not tidy because one column contains both year and sex
dt <- separate(dt, col = "filename", into = c("year", "sex"), extra = "drop")
head(dt)
```

```
##    year sex    name  percent
## 1: 1880 boy    John 0.081541
## 2: 1880 boy William 0.080511
## 3: 1880 boy   James 0.050057
## 4: 1880 boy Charles 0.045167
## 5: 1880 boy  George 0.043292
## 6: 1880 boy   Frank 0.027380
```

Section 04 - Merge Warm Up

Prepare two tables by running the following code:

```
mtcars_dt <- as.data.table(mtcars)
mtcars_dt[, carname := rownames(mtcars)]

dt1 <- mtcars_dt[5:25,.(carname, mpg, cyl)]
dt2 <- mtcars_dt[1:10, .(carname, gear)]
```

1. How long is the inner merge of dt1 and dt2?

```
inner_dt <- merge(dt1, dt2, by='carname')
inner_dt
```

```
##              carname  mpg cyl gear
## 1:        Duster 360 14.3   8    3
## 2: Hornet Sportabout 18.7   8    3
## 3:          Merc 230 22.8   4    4
## 4:         Merc 240D 24.4   4    4
## 5:          Merc 280 19.2   6    4
## 6:          Valiant 18.1   6    3
```

```
inner_dt[, .N]
```

```
## [1] 6
```

2. How long is the left merge of dt1 and dt2?

```
left_dt <- merge(dt1, dt2, by='carname', all.x = T)
left_dt
```

```
##                 carname  mpg cyl gear
##  1:         AMC Javelin 15.2   8   NA
##  2:  Cadillac Fleetwood 10.4   8   NA
##  3:          Camaro Z28 13.3   8   NA
##  4:   Chrysler Imperial 14.7   8   NA
##  5:    Dodge Challenger 15.5   8   NA
##  6:          Duster 360 14.3   8    3
##  7:            Fiat 128 32.4   4   NA
##  8:         Honda Civic 30.4   4   NA
##  9:   Hornet Sportabout 18.7   8    3
## 10: Lincoln Continental 10.4   8   NA
## 11:            Merc 230 22.8   4    4
## 12:           Merc 240D 24.4   4    4
## 13:            Merc 280 19.2   6    4
## 14:           Merc 280C 17.8   6   NA
## 15:          Merc 450SE 16.4   8   NA
## 16:          Merc 450SL 17.3   8   NA
## 17:         Merc 450SLC 15.2   8   NA
## 18:    Pontiac Firebird 19.2   8   NA
## 19:      Toyota Corolla 33.9   4   NA
## 20:       Toyota Corona 21.5   4   NA
## 21:             Valiant 18.1   6    3
##                 carname  mpg cyl gear
```

```
left_dt[, .N]
```

```
## [1] 21
```

3. How long is the outer merge of dt1 and dt2?

```
outer_dt <- merge(dt1, dt2, by='carname', all = T)
outer_dt
```

```
##                 carname  mpg cyl gear
##  1:         AMC Javelin 15.2   8   NA
##  2:  Cadillac Fleetwood 10.4   8   NA
##  3:          Camaro Z28 13.3   8   NA
##  4:   Chrysler Imperial 14.7   8   NA
##  5:          Datsun 710   NA  NA    4
##  6:    Dodge Challenger 15.5   8   NA
##  7:          Duster 360 14.3   8    3
##  8:            Fiat 128 32.4   4   NA
##  9:         Honda Civic 30.4   4   NA
## 10:      Hornet 4 Drive   NA  NA    3
## 11:   Hornet Sportabout 18.7   8    3
## 12: Lincoln Continental 10.4   8   NA
## 13:           Mazda RX4   NA  NA    4
## 14:       Mazda RX4 Wag   NA  NA    4
## 15:            Merc 230 22.8   4    4
```

```
## 16:           Merc 240D 24.4   4    4
## 17:            Merc 280 19.2   6    4
## 18:           Merc 280C 17.8   6   NA
## 19:          Merc 450SE 16.4   8   NA
## 20:          Merc 450SL 17.3   8   NA
## 21:         Merc 450SLC 15.2   8   NA
## 22:    Pontiac Firebird 19.2   8   NA
## 23:      Toyota Corolla 33.9   4   NA
## 24:       Toyota Corona 21.5   4   NA
## 25:             Valiant 18.1   6    3
##                carname  mpg cyl gear
```

```
outer_dt[, .N]
```

```
## [1] 25
```

Section 05 - Small case-study: cleaning up a gene-expression dataset in yeast

In this section, we will read and clean up the data from the paper:

- Gagneur, Julien, et al. "Genotype-environment interactions reveal causal pathways that mediate genetic effects on phenotype." PLoS Genet 9.9 (2013): e1003803. https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1003803

1. Read in the two files in the folder `eqtl`.

The first file contains the genotypes of yeast strains and a strain identifyer. The second file contains information on how quickly each strain grows in different growth media.

```
gt <- fread('extdata/eqtl/genotype.txt')
dim(gt)
```

```
## [1]  158 1001
```

```
head(gt[,1:5])
```

```
##     strain        mrk_1       mrk_14       mrk_27       mrk_40
## 1: seg_01B   Lab strain   Lab strain   Lab strain   Lab strain
## 2: seg_01C Wild isolate Wild isolate Wild isolate Wild isolate
## 3: seg_01D   Lab strain   Lab strain   Lab strain   Lab strain
## 4: seg_02B   Lab strain   Lab strain   Lab strain   Lab strain
## 5: seg_02C Wild isolate Wild isolate Wild isolate   Lab strain
## 6: seg_02D   Lab strain   Lab strain   Lab strain Wild isolate
```

```
growth <- fread('extdata/eqtl/growth.txt')
head(growth)
```

```
##     strain      YPD   YPD_BPS YPD_Rapa      YPE   YPMalt
## 1: seg_01B 12.60399 10.460795 2.500311 5.265698 6.720447
## 2: seg_01C 10.79114 11.632019       NA 5.365259 7.429273
```

```
## 3: seg_01D 12.81727 10.423287 3.142154 5.577932 6.905589
## 4: seg_02B 10.29921  9.103611 4.314388 3.257843 4.924324
## 5: seg_02C 11.13278  9.263100 3.548543 3.815689 4.413402
## 6: seg_02D 13.91084 11.750178       NA 5.672890 7.926200
```

2. Come up with a strategy, how you can transform the two tables shown above into the single table shown below.

```
head(dt)
```

```
##     strain media growth_rate marker         gt
## 1: seg_01B   YPD    12.60399  mrk_1 Lab strain
## 2: seg_01B   YPD    12.60399 mrk_14 Lab strain
## 3: seg_01B   YPD    12.60399 mrk_27 Lab strain
## 4: seg_01B   YPD    12.60399 mrk_40 Lab strain
## 5: seg_01B   YPD    12.60399 mrk_54 Lab strain
## 6: seg_01B   YPD    12.60399 mrk_67 Lab strain
```

```
summary(dt)
```

```
##      strain            media         growth_rate        marker
##  seg_01B:  5000   YPD     :158000   Min.   : 1.57   mrk_1  :   790
##  seg_01C:  5000   YPD_BPS :158000   1st Qu.: 4.55   mrk_14 :   790
##  seg_01D:  5000   YPD_Rapa:158000   Median : 6.93   mrk_27 :   790
##  seg_02B:  5000   YPE     :158000   Mean   : 7.60   mrk_40 :   790
##  seg_02C:  5000   YPMalt  :158000   3rd Qu.:10.70   mrk_54 :   790
##  seg_02D:  5000                     Max.   :16.27   mrk_67 :   790
##  (Other):760000                     NA's   :42000   (Other):785260
##             gt
##  Lab strain  :398145
##  Wild isolate:391855
##
##
##
##
##
```

```
# - melt each table.
# - merge them by the strain column
# - and convert the character columns into factors.
```

3. Write code that implements you strategy to transform the two tables into the one shown above.

```
gt <- fread('extdata/eqtl/genotype.txt')
dim(gt)
```

```
## [1]  158 1001
```

```
head(gt[,1:5])
```

```
##     strain        mrk_1       mrk_14       mrk_27       mrk_40
## 1: seg_01B   Lab strain   Lab strain   Lab strain   Lab strain
## 2: seg_01C Wild isolate Wild isolate Wild isolate Wild isolate
## 3: seg_01D   Lab strain   Lab strain   Lab strain   Lab strain
## 4: seg_02B   Lab strain   Lab strain   Lab strain   Lab strain
## 5: seg_02C Wild isolate Wild isolate Wild isolate   Lab strain
## 6: seg_02D   Lab strain   Lab strain   Lab strain Wild isolate
```

```
growth <- fread('extdata/eqtl/growth.txt')
```

```
# melt both datasets
gt <- melt(gt, id.vars = 'strain', value.name = 'gt', variable.name='marker')
growth <- melt(growth, id.vars = 'strain', variable.name = 'media', value.name = 'growth_rate')

# merge them by strain. As every row gets merged with every row we need to allow cartesian.
dt <- merge(growth, gt, by='strain', allow.cartesian = TRUE)

# convert the categorical entries to factors.
# (Measuring the object size before and after shows that factors require less storage space.)

#object.size(dt)
dt[,gt:= as.factor(gt)]
dt[,strain:= as.factor(strain)]
dt[,marker:= as.factor(marker)]
#object.size(dt)

head(dt)
```

```
##     strain media growth_rate marker         gt
## 1: seg_01B   YPD    12.60399  mrk_1 Lab strain
## 2: seg_01B   YPD    12.60399 mrk_14 Lab strain
## 3: seg_01B   YPD    12.60399 mrk_27 Lab strain
## 4: seg_01B   YPD    12.60399 mrk_40 Lab strain
## 5: seg_01B   YPD    12.60399 mrk_54 Lab strain
## 6: seg_01B   YPD    12.60399 mrk_67 Lab strain
```

```
summary(dt)
```

```
##      strain            media         growth_rate        marker
##  seg_01B:  5000   YPD     :158000   Min.   : 1.57   mrk_1  :   790
##  seg_01C:  5000   YPD_BPS :158000   1st Qu.: 4.55   mrk_14 :   790
##  seg_01D:  5000   YPD_Rapa:158000   Median : 6.93   mrk_27 :   790
##  seg_02B:  5000   YPE     :158000   Mean   : 7.60   mrk_40 :   790
##  seg_02C:  5000   YPMalt  :158000   3rd Qu.:10.70   mrk_54 :   790
##  seg_02D:  5000                     Max.   :16.27   mrk_67 :   790
##  (Other):760000                     NA's   :42000   (Other):785260
##             gt
##  Lab strain  :398145
##  Wild isolate:391855
##
##
##
##
##
```

4. (Optional) When you are done run the following line of code and observe the result:

```
library(ggplot2)
ggplot(dt[marker %in% c('mrk_5211', 'mrk_1653')], aes(marker, growth_rate, color=gt)) +
    geom_boxplot() + facet_wrap(~media)
```

```
# This example ilustrates that tidy formats make it simple to plot (complicated) relationships.
# Next week will cover how to create such plots.
```